Evaluation of the SPLICE Algorithm on the Aurora2 Database

Jasha Droppo, Li Deng, and Alex Acero

Microsoft Research, One Microsoft Way, Redmond, WA
{jdroppo,deng,acero}@microsoft.com

Abstract

This paper describes recent improvements to SPLICE, Stereo-based Piecewise Linear Compensation for Environments, which produces an estimate of cepstrum of undistorted speech given the observed cepstrum of distorted speech. For distributed speech recognition applications, SPLICE can be placed at the server, thus limiting the processing that would take place at the client. We evaluated this algorithm on the Aurora2 task, which consists of digit sequences within the TIDigits database that have been digitally corrupted by passing them through a linear filter and/or by adding different types of realistic noises at SNRs ranging from 20dB to -5dB. On set A data, for which matched training data is available, we achieved a 66% decrease in word error rate over the baseline system with clean models. This preliminary result is of practical significance because in a server implementation, new noise conditions can be added as they are identified once the service is running.

1. Introduction

There has been a great deal of interest recently in standardizing distributed speech recognition applications in which the user can have either a plain phone or a smart phone and speech recognition is done at a centralized server. Because of bandwidth limitations, one possibility is to have a cellular phone use a standard codec to transmit the speech to the server, which decompresses it and recognizes it. Since ASR systems only need some features of the speech signal, such as mel-cepstrum, more bandwidth can be saved by transmitting only those features. ETSI has been accepting proposals for Aurora [1], an effort to standardize a front-end for distributed speech recognition applications that offers low bitrate and is robust to noise and channel distortions.

The SPLICE enhancement technique described in this paper is front-end agnostic. That is, it makes no assumptions on the structure and processing of the front end and merely tries to undo whatever corruption it is shown during training. In a distributed speech recognition system, the SPLICE may either be applied within the front end on the client device, or on the server. Implementation on the server has several advantages. Computational complexity becomes less of an issue, and continuing improvements can be made that benefit devices already deployed in the field.

SPLICE is a frame-based bias removal algorithm for cepstrum enhancement under additive noise distortion, channel distortion, or a combination of the two. In [2] we reported the approximate MAP formulation of the algorithm, and more recently [3][4] described the MMSE formulation of the algorithm with a much wider range of naturally recorded noise including both artificially mixed speech and noise and naturally recorded noisy speech. In this paper, we report some new developments of the algorithm that take into account the inter-frame dynamic mapping between the clean and distorted speech by using temporal smoothing, and present full sets of evaluation results for AURORA2 digit-sequence recognition.

The SPLICE algorithm assumes no explicit noise model, and the noise characteristics are embedded in the piecewise linear mapping between the "stereo'' clean and distorted speech cepstral vectors. The piecewise linearity is intended to approximate the true nonlinear relationship between the two. The nonlinearity between the cepstral vectors of clean and distorted (including additive noise) cepstra arises due to the use of the logarithm in computing the cepstra. Because of the use of the stereo training data that provide accurate estimates of the bias or correction vectors without the need for an explicit noise model, the SPLICE algorithm is potentially able to effectively handle a wide range of difficult distortions, including nonstationary distortion, joint additive and convolutional distortion, and even nonlinear distortion of the original time-series. A key requirement for the success of the current version of the SPLICE is that the distortion conditions under which the correction vectors are learned from the stereo data are similar to those that corrupt the test data. Future development of the algorithm will relax this requirement by employing linear combinations of the distortion conditions and by sequentially adapting the distortion conditions.

This organization of this paper is as follows. In Section 2, we give a brief review of the basic SPLICE algorithm. The extension of the basic SPLICE to its dynamic, temporally smoothed version is presented in Section 3. A blind equalization method needed to enhance the performance of SPLICE is described in Section 4. Full results of digit-sequence recognition for AURORA2 are presented and discussed in Section 5.

2. A Review of SPLICE

Given the general model of distortion from a clean cepstral vector, **x**, into a noisy one, **y**, we describe the probabilistic formulation of the basic (frame independent) version of the SPLICE algorithm below.

2.1. A Model of Speech and its Degradation

The first assumption is that the noisy speech cepstral vector follows the distribution of mixture of Gaussians:

$$p(\mathbf{y}) = \sum_s p(\mathbf{y} \mid s)\, p(s), \text{ where}$$
$$p(\mathbf{y} \mid s) = N(\mathbf{y}; \boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s). \quad (1)$$

The discrete state variable s denotes the discrete random variable taking the values 1, 2, …, N, one for each region over which the piecewise linear approximation between the clean cepstral vector **x** and distorted cepstral vector is made. One distribution $p(\mathbf{y})$, is trained for each separate distortion condition (not indexed for clarity), and can be thought as a

"codebook" with a total of *N* codewords (means) and their variances.

The second assumption made by the SPLICE is that the conditional probability density function (PDF) for the clean vector **x** given the noisy speech vector, **y**, and the region index, *s*, is Gaussian whose mean vector is a linear transformation of the noisy speech vector **y**. In this paper, we take a simplified form of this linear transformation by making the rotation matrix to be the identity matrix, leaving only the bias or correction vector. Thus, the conditional PDF is assumed to have the form,

$$p(\mathbf{x}\,|\,\mathbf{y},s) = N(\mathbf{x};\mathbf{y}+\mathbf{r}_s,\boldsymbol{\Gamma}_s)\,. \tag{2}$$

2.2. Cepstral Enhancement

One significant advantage of the above two basic assumptions made in the SPLICE is the inherent simplicity in deriving and implementing the rigorous MMSE estimate of clean speech cepstral vectors from their distorted counterparts. The MMSE is the following conditional expectation of clean speech vector given the observed noisy speech:

$$\hat{\mathbf{x}}_{MMSE} = E_x[\mathbf{x}\,|\,\mathbf{y}] = \sum_s p(s\,|\,\mathbf{y})E_{\mathbf{x}}[\mathbf{x}\,|\,\mathbf{y},s]. \tag{3}$$

Using Eq. (2), it is clear that:

$$E_x[\mathbf{x}\,|\,\mathbf{y},s] = \mathbf{y}+\mathbf{r}_s\,, \tag{4}$$

which, inserted into Eq. (3), results in

$$\hat{\mathbf{x}}_{MMSE} = \mathbf{y} + \sum_s p(s\,|\,\mathbf{y})\mathbf{r}_s \tag{5}$$

so that the MMSE estimate of **x** is the noisy speech vector corrected by a linear weighted sum of all codeword-dependent bias vectors.

A faster implementation can be achieved by approximating the weights $p(s\,|\,\mathbf{y})$ according to

$$\hat{p}(s\,|\,\mathbf{y}) \cong \begin{cases} 1 & s = \arg\max_s p(s\,|\,\mathbf{y}) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

so that this approximation turns the MMSE estimate to the approximate MAP estimate [2] that consists of two sequential steps of operation. First, finding optimal codeword *s* using the VQ codebook based on the parameters ($\boldsymbol{\mu}_s,\boldsymbol{\Sigma}_s$), and then adding the codeword-dependent vector $\mathbf{r}_s$ to the noisy speech vector. We have found empirically that the above VQ approximation does not appreciably affect recognition accuracy. All results presented in this paper use this approximation.

2.3. SPLICE Training

Since the noisy speech PDF $p(\mathbf{y})$ is assumed to be a mixture of Gaussians, the standard EM algorithm can be used to train $\boldsymbol{\mu}_s$ and $\boldsymbol{\Sigma}_s$ on noisy speech. Initial values of the parameters are determined by a VQ clustering algorithm.

If stereo data is available, the parameters $\mathbf{r}_s$ of the conditional PDF $p(\mathbf{x}\,|\,\mathbf{y},s)$ can be trained using the maximum likelihood criterion:

$$\mathbf{r}_s = \frac{\sum_n p(s\,|\,\mathbf{y}_n)(\mathbf{x}_n-\mathbf{y}_n)}{\sum_n p(s\,|\,\mathbf{y}_n)}\,, \text{ where} \tag{7}$$

$$p(s\,|\,\mathbf{y}_n) = \frac{p(\mathbf{y}_n\,|\,s)\,p(s)}{\sum_s p(\mathbf{y}_n\,|\,s)} \tag{8}$$

where this training procedure requires a set of stereo (two channel) data. One channel contains the clean utterance, and the other contains the same utterance with distortion, where the distortion represented by the correction vectors is estimated above. The two-channel data can be collected, for example, by simultaneously recording utterances with one close-talk and one far-field microphone.

For the Aurora work reported in this paper, the SPLICE parameters were trained using identical utterances from the clean training set and the multi-style training set. This effectively tunes our cepstral enhancement parameters on the noise types from set A, keeping sets B and C as unseen conditions.

Note that the correction vectors $\mathbf{r}_s$ can also be estimated without the need of stereo data, at the expense of modest loss in accuracy [5].

2.4. Environmental Model Selection

The SPLICE algorithm described so far requires that the mixture of Gaussians for the noisy speech be conditioned on a specific noise type and level. To satisfy this requirement, we developed an effective on-line environmental selection method, which has been described in detail in [4].

We apply this method to the AURORA2 evaluation as follows. Twenty separate mixture models are trained, one for each of the combinations of noise type and level in the multicondition training set. The on-line decision for selecting the environmental model *e* is made by first producing a local estimate of $p(\mathbf{y}_i\,|\,e)$ and then smoothing it over time.

In a server-based application, new environment codebooks could be created as more incoming speech is received if $p(\mathbf{y}_i\,|\,e)$ is below a threshold, so that little mismatch occurs. Although we do not have stereo data in this case, a stereo database can be created by digitally adding a small amount of noise present in the unseen environment to a larger clean database [4]. If channel distortion is also present and can be estimated, the clean speech can also be filtered. We did not do that for the Aurora evaluation because of the way it was defined, but it could bridge the performance gap between seen and unseen environments.

3. Dynamic SPLICE

In this section, we present a new version of SPLICE that not only minimizes the static deviation from the clean to noisy cepstral vectors (as in the basic version of the SPLICE

described in Section 2), but also seeks to minimize the deviation between the delta parameters.

The basic SPLICE (optimally) processes each frame of noisy speech independently. An obvious extension is to jointly process a segment of frames. In this way, although the deviation from the clean to noisy speech cepstra for an individual frame may be undesirably greater than that achieved by the basic, static SPLICE, the global deviation that takes into account the differential frames and the whole segment of frames can be reduced compared with the basic SPLICE.

We have implemented the above idea of dynamic SPLICE by temporally smoothing the bias vectors obtained from the basic, static SPLICE described in Section 2. This is an empirical way of implementing the rigorous solution which would use a more realistic model for the time-evolution of the clean speech dynamics. Using the discrete state, we would model $p(\mathbf{x}_n \mid \mathbf{y}_n, s_n, s_{n-1})$, or using the continuous clean speech vector estimate we would model $p(\mathbf{x}_n \mid \mathbf{y}_n, s_n, \mathbf{x}_{n-1})$.

An efficient way to implement an approximate dynamic SPLICE, as is used in the current AURORA2 evaluation, is to independently time-filter each component of the cepstral bias vector $\mathbf{r}_{s_n}$. We have achieved significant performance gains using this efficient heuristic implementation.

In our specific implementation, we used a simple zero-phase, non-causal, IIR filter to smooth the cepstral bias vectors. This filter has a low-pass characteristic, with the system transfer function of

$$H(z) = \frac{-0.5}{\left(z^{-1} - 0.5\right)\left(z - 2\right)}. \quad (9)$$

This transfer function is the result of defining an objective function of the summation of the static and dynamic deviations from clean speech to noisy speech vectors. The optimal solution that minimizes this objective function is of the form of Eq. (9), where the constants are functions of the variances of our model. In practice, using Eq. (9), instead of the exact solution, produces similar results at a lower computational cost.

4. Blind Equalization

In principle, when the training data for the SPLICE contain similar convolutional distortions to those in the test data, the algorithm described above can effectively remove that distortion. However, for data in set C, the convolutional distortion is unknown, so the stereo data needed for Eq. (7) is unavailable. This requires a modification of the existing SPLICE that explicitly takes into account the convolutional distortion. Blind equalization described here is an effective way of achieving this.

We modify our probabilistic model to include an unknown vector common to all codewords, **h**, which accounts for the fixed channel distortion. The modified *pdf* for noisy speech becomes

$$p_{\mathbf{y}|\mathbf{h},s}(\mathbf{y} \mid \mathbf{h}, s) = p_{\mathbf{y}|s}(\mathbf{y} - \mathbf{h} \mid s). \quad (10)$$

Estimation of the unknown **h** follows from maximizing the probability of the observed noisy speech, based on Eq. (10). We have implemented this by alternating maximizations. First, we find **h** that maximizes (10), holding the discrete state sequence (s_n) constant. Then, we find a new state sequence while holding **h** constant. We have found that convergence is reached in about five iterations if done in batch mode. An online version is also possible.

This blind equalization technique increases the average word error rate on set A by just 3.9% relative, while bringing the average word error rate on set C down by 20.5% relative.

We also compared this technique to a much simpler CMN implementation, in which the mean cepstrum of each file was subtracted from that file. This simple CMN technique led to an average word error rate reduction of 6.6% relative to SPLICE alone on the entire Aurora2 task. Our blind equalization discussed here lead to an average word error rate reduction of 7.2%.

5. Experimental Results Using AURORA2

The speech recognition results reported in this paper are produced by the reference Aurora front-end version 2.0, using c_0 instead of log energy, and modified to use power spectral density instead of magnitude spectrum in its computations. We found this configuration to be slightly superior to the default.

Table 1 is a summary of the full results for the SPLICE on the Aurora2 corpus. The SPLICE parameters were trained using the clean and multi-style data corresponding to noises from Set A. The noisy speech model consisted of a mixture of 256 Gaussians with diagonal covariance matrices, though we have observed improved accuracy for some nonstationary noise types with more Gaussians. The bias vectors were smoothed according to Eq. (9). We now discuss these results.

Set A: Since the SPLICE parameters were trained on fixed noise conditions that are included in Set A, this set exhibits the best performance among all the three sets. While Set A only contains four types of noises, we have experimented up to fourteen noise types, giving similarly good results on a Wall Street Journal task. (We call the algorithm applied to this experimental setup as in-task or in-condition SPLICE in [3].)

Set B: To examine the SPLICE's ability to perform in unseen noise conditions, we applied the SPLICE parameters developed for set A, without modification, to enhance the cepstra in sets B and C. This experimental setup does not allow the in-condition SPLICE to apply. We called this more difficult experimental setup, where the noisy condition in the stereo training data is unseen in the test data, the cross-task or cross-condition SPLICE in [3]. From the results in Table 1, we observe reasonable performance improvement over the baseline, consistent for all noise conditions, when using the clean acoustic model. This improvement is less than that achieved for set A. This indicates that the bias vectors learned from set A's stereo data may not be representative of those required to transform the noisy speech to clean speech in set B.

Set C: Among the two noise types in Set C, Subway condition has been included when training the SPLICE parameters, and Street condition has not. And this test set also includes a convolutional distortion not seen the training set of Set A. For the Subway noise type, we have the in-task condition and the word accuracy is almost as good as for the unfiltered test data from Set A. Use of the in-task SPLICE

Table 1: SPLICE results on the Aurora2 database.

Aurora 2 Multicondition Training - Results

	A					B					C				
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	Overall	Percentage Improvement
Clean	98.74	98.64	98.48	98.77	98.66	98.74	98.64	98.48	98.77	98.66	98.80	98.64	98.72	**98.67**	**9.78%**
20 dB	98.31	98.46	98.18	98.36	98.33	97.70	97.19	96.30	97.90	97.27	98.16	97.13	97.65	**97.77**	**15.00%**
15 dB	97.36	97.10	97.76	97.35	97.39	96.16	95.28	92.69	94.51	94.66	97.36	95.44	96.40	**96.10**	**-4.12%**
10 dB	95.61	95.01	96.12	95.12	95.47	93.31	90.30	88.79	89.48	90.47	95.49	90.78	93.14	**93.00**	**-11.88%**
5 dB	92.54	86.28	90.87	89.54	89.81	81.67	73.67	78.74	78.96	78.26	91.03	75.88	83.46	**83.92**	**-9.92%**
0 dB	79.43	59.37	74.20	74.48	71.87	53.48	43.62	48.79	51.96	49.46	76.48	47.88	62.18	**60.97**	**2.85%**
-5dB	48.48	18.83	34.09	44.89	36.57	9.70	14.12	4.44	13.39	10.41	40.22	17.20	28.71	**24.54**	**-0.50%**
Average	92.65	87.24	91.43	90.97	**90.57**	84.46	80.01	81.06	82.56	**82.03**	91.70	81.42	**86.56**	**86.35**	
	34.63%	**-5.86%**	**36.38%**	**24.56%**	**22.63%**	**-6.35%**	**-54.20%**	**-53.24%**	**-16.33%**	**-30.92%**	**50.49%**	**-18.41%**	**17.17%**		**-0.28%**

Aurora 2 Clean Training - Results

	A					B					C				
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	Overall	Percentage Improvement
Clean	98.89	99.03	98.96	99.20	99.02	98.89	99.03	98.96	99.20	99.02	99.02	99.03	99.03	**99.02**	**-0.96%**
20 dB	97.97	98.46	98.39	98.27	98.27	98.34	97.19	97.46	98.36	97.84	97.82	97.22	97.52	**97.95**	**57.43%**
15 dB	96.50	96.98	97.35	96.39	96.81	96.44	94.74	94.78	95.74	95.43	96.53	94.74	95.64	**96.02**	**67.98%**
10 dB	93.25	93.89	94.24	92.53	93.48	92.94	87.70	90.22	90.00	90.22	92.60	88.18	90.39	**91.56**	**73.41%**
5 dB	86.95	82.16	86.22	84.36	84.92	81.55	68.26	77.48	76.67	75.99	85.78	71.01	78.40	**80.04**	**66.98%**
0 dB	69.48	49.24	62.39	64.67	61.45	50.94	37.42	45.69	46.47	45.13	65.49	42.65	54.07	**53.44**	**43.98%**
-5dB	37.27	17.05	26.25	33.23	28.45	11.58	13.75	6.56	12.96	11.21	30.00	16.17	23.09	**20.48**	**13.13%**
Average	88.83	84.15	87.72	87.24	**86.98**	84.04	77.06	81.13	81.45	**80.92**	87.64	78.76	**83.20**	**83.80**	
	63.39%	**68.36%**	**68.83%**	**63.14%**	**66.33%**	**66.34%**	**40.39%**	**59.63%**	**58.19%**	**56.88%**	**63.48%**	**37.31%**	**50.39%**		**59.44%**

together with blind equalization successfully brings the enhanced cepstra close to the clean cepstra. For the Street noise type, mismatch of the noise types accounts for the significantly lower performance.

6. Summary and Discussion

The SPLICE algorithm, as described in this paper, is an efficient algorithm that can be run either on the client or the server in a distributed speech recognition system. It models cepstra of noisy speech as a mixture of Gaussians. We can leverage this model to identify the type of corruption currently being encountered, and to compensate for an unknown linear filter. By incorporating the dynamic SPLICE modification, word error rate decreases universally across both seen and unseen distortion conditions.

We achieve significant improvements on Aurora set A, where we are able to directly use our current training algorithms, which require stereo training data. SPLICE also improves, to a lesser degree, word error rate in unseen noise conditions. To obtain better improvements in set B, we would need to augment the set of different noise types in training the correction vectors. In a server implementation, as new noise conditions are identified, they can be added to the set of distortions that SPLICE is capable of removing.

The most immediate obstacle to the full success of the current algorithm is the cross-condition task when noise type/level mismatch occurs. This has been clearly demonstrated in the Set B results. Several ways of overcoming this obstacle are currently under investigation. These include:

1. Tracking the on-line noise and adapt the bias vectors and mixing weights.
2. Train the bias vectors in SPLICE with many more noise types in order to cover a wider range of noisy environments expected to encounter.
3. Using a combination of the existing, limited noisy environments to improve generalization to a possibly unseen environment, and
4. Relaxation of our need for stereo training data. (Similar algorithms have removed these restrictions, with some success [5].)

We are also investigating real-time implementations of parametric methods [6].

7. References

[1] H. G. Hirsch and D. Pearce, “The AURORA Experimental Framework for the Performance Evaluations of Speech Recognition Systems under Noisy Conditions,” ISCA ITRW ASR2000 “Automatic Speech Recognition: Challenges for the Next Millennium,” Paris, France, September 18-20, 2000.

[2] L. Deng, A. Acero, M. Plumpe and X. Huang, “Large Vocabulary Continuous Speech Recognition under Adverse Conditions,” *Proc. of the ICSLP*, Beijing, October 2000, Vol. 3, pp. 806-809.

[3] L. Deng, A. Acero, L. Jiang, J. Droppo and X. Huang. “High-Performance Robust Speech Recognition using Stereo Training Data,” in *Int. Conf. On Acoustics, Speech and Signal Processing*, Salt Lake City, Utah, May 2001.

[4] J. Droppo, A. Acero and L. Deng, “Efficient On-Line Acoustic Environment Estimation for FCDCN in a Continuous Speech Recognition System,” in *Int. Conf. On Acoustics, Speech and Signal Processing*, Salt Lake City, Utah, May 2001.

[5] P. Moreno, “Speech Recognition in Noisy Environments,” PhD thesis, Carnegie Mellon University, 1996.

[6] B. Frey, L. Deng, A. Acero and T. Kristjansson. “ALGONQUIN: Iterating Laplace’s Method to Remove Multiple Types of Noise and Channel Distortion from Log-Spectra in Robust Speech Recognition”. Submitted to EuroSpeech 2001.